

Bakalářské zkoušky (příklady otázek)

jaro 2018

1 Automaty a gramatiky (3 body)

1. Definujte deterministický a nedeterministický konečný automat.
2. Jak se liší jazyky přijímané těmito dvěma druhy automatů?
3. Dokažte, že následující jazyk je regulární: obsahuje všechna slova nad abecedou $\{a, k, t\}$, v nichž se jako podslovo vyskytuje *katka* nebo *ataka* (mohou i obě).

2 Algoritmy a datové struktury: třídy složitosti (3 body)

1. Definujte třídy složitosti **P** a **NP**.
2. Definujte převoditelnost problémů.
3. Definujte, co znamená, že problém je **NP**-úplný.
4. Převedte následující problém na SAT (splnitelnost booleovských formulí v konjunktivní normální formě): Jsou dány množiny X, Y, Z a množina trojic $T \subseteq X \times Y \times Z$. Lze ze zadaných trojic vybrat podmnožinu $T' \subseteq T$ takovou, aby se v ní každý prvek množin X, Y a Z vyskytl právě jednou?

3 Databáze (3 body)

Součástí studijního informačního systému jsou mj. tabulky

```
STUDENT(ST_ID, JMENO, PRIJMENI)
```

```
UCITEL(UC_ID, JMENO, PRIJMENI)
```

```
PREDMET(PR_ID, NAZEV, UC_ID)
```

```
ZAPIS(SKR, ST_ID, PR_ID, UC_ID)
```

ST_ID , UC_ID a PR_ID jsou unikátní kódy studentů, učitelů a předmětů, SKR je školní rok. Zápis studenta na předmět se provádí SQL příkazem

```
INSERT INTO ZAPIS(SKR, ST_ID, PR_ID, UC_ID)
```

```
VALUES( :SKR, :ST_ID, :PR_ID, (SELECT UC_ID FROM PREDMET WHERE PR_ID = :PR_ID))
```

kde $:SKR$, $:ST_ID$ a $:PR_ID$ jsou zadané parametry příkazu. Účelem tabulky ZAPIS je trvale zaznamenat skutečnost, že v daném školním roce byl student zapsán na daný předmět, který přitom vyučoval uvedený vyučující.

1. Dá se z výše uvedeného určit, zda může mít předmět více vyučujících současně? Uveďte stručně relevantní argumenty pro/proti.
2. Napište SQL příkaz nebo příkazy realizující změnu vyučujícího daného předmětu v době, kdy jsou již někteří studenti pro aktuální školní rok zapsáni. Změna nesmí zasáhnout zápisy z předchozích roků.
3. Kde je v uvedeném schématu redundance a jakou úpravu je vhodné provést pro její odstranění? Která normální forma byla redundancí porušena?

4 Programovací jazyky (3 body)

Uvažujte systém pro vyhledávání v jízdních řádech, který je rozdělen na dvě vrstvy - nižší vrstva realizuje databázi stanic a spojů, vyšší vrstva implementuje algoritmus vyhledávání nejrychlejšího spojení.

Předpokládejte, že vyšší vrstva používá nějaký vhodný jednoduchý algoritmus (např. prohledávání do šířky) pro vyhledávání spojení. Rozmyslete si, jak by tento algoritmus fungoval a jaké funkce od nižší vrstvy by požadoval.

Na základě toho navrhnete vnější rozhraní nižší (databázové) vrstvy, které bude algoritmus vyšší vrstvy využívat.

Vaším úkolem je pouze návrh tohoto rozhraní, např. ve formě seznamu metod s popisem parametrů a návratových hodnot, a definice složitějších typů použitých v tomto rozhraní. Způsob uložení dat v nižší vrstvě řešte pouze tehdy, pokud se projevuje v navrženém rozhraní.

Detaily jako závislost jízdnic řádů na dni v týdnu neřešte. Inicializaci ani aktualizaci databáze neřešte a předpokládejte, že v průběhu vyhledávání k aktualizaci nedochází.

Pro řešení úlohy si vyberte jeden z jazyků Java, C++, C#. Při hodnocení odpovědi nebudou uvažovány drobné syntaktické chyby, ale obecně by použité konstrukce měly odpovídat zvolenému programovacímu jazyku. Věnujte pozornost otázkám, která data (objekty) jednotlivé metody vytvářejí/kopírují a která data pouze zpřístupňují - zbytečná kopírování objemných dat budou při hodnocení penalizována.

5 Architektura počítačů a operačních systémů (3 body)

Uvažujte architekturu procesoru s podporou stránkování a délkou virtuální a fyzické adresy 32 bitů. Pro jednoduchost předpokládejte, že k překladu adres procesor používá jednoúrovňovou stránkovací tabulku. Velikost stránek je 4 KiB.

1. Kolik položek bude mít stránkovací tabulka a kolik paměti bude tato tabulka zabírat pro každý spuštěný proces? Položka stránkovací tabulky musí obsahovat všechny nutné informace a musí být pro procesor efektivně přístupná (maximálně 1 čtení z paměti).
2. Přístup na virtuální adresu 0x00020748 byl přeložen na fyzickou adresu 0x882F9748. Dále předpokládejte, že stránkovací tabulka procesu je uložena v souvislém bloku fyzické paměti od adresy 0x00800000. Na jaké fyzické adrese leží položka stránkovací tabulky použitá při překladu a jaký je její obsah? Uveďte hodnoty a význam všech (skupin) bitů záznamu.
3. Předpokládejte, že při přístupu procesu na virtuální adresu 0x00020748 nedošlo k výpadku stránky. Bezprostředně poté proces přečetl 8-bajtové slovo z adresy 0x00020FF8. Můžete usoudit, zda dojde k výpadku stránky u druhého přístupu? Odpověď zdůvodněte.

V odpovědích podle potřeby používejte hexadecimální (případně binární) zápis čísel.

6 Základy sítí (3 body)

Váš počítač právě nastartoval. Po přihlášení uživatele a spuštění webového prohlížeče napíšete do adresového řádku:

`http://is.cuni.cz/studium/login.php`

1. Aby se prohlížeč mohl připojit k serveru „is.cuni.cz“, musí nejprve převést jeho jméno na IP adresu, k čemuž typicky použije knihovni funkci. Rámcově popište, jak tento převod na úrovni síťové komunikace probíhá: jaké informace musí mít klientský počítač k dispozici, aby mohl takový překlad vůbec iniciovat, na jaké záznamy se bude ptát a jaké může dostat odpovědi?
2. Poté, co prohlížeč zjistil adresu serveru, potřebuje získat obsah zdroje „/studium/login.php“ ze serveru. Popište jak proběhne komunikace se serverem. Na úrovni transportního protokolu uveďte, jakým protokolem a s kým bude klientský počítač komunikovat, a na úrovni aplikačního protokolu uveďte, jak bude vypadat dotaz klienta a odpověď serveru.
3. Prohlížeč ze serveru obdržel dokument, v němž je následující fragment:

```
<form name="flogin" action="./verif.php" method="POST">
  <p>Login: <input type="text" size="30" name="login" value="">
  <p>Heslo: <input type="password" size="40" name="heslo">
  <p><input type="submit" name="all" value="Login">
</form>
```

Popište komunikaci prohlížeče a serveru na úrovni aplikačního protokolu poté, co vyplníte údaje ve formuláři a kliknete na tlačítko `Login`. Uveďte, jak bude vypadat dotaz klienta a odpověď serveru.

7 Překladače (3 body)

Tato gramatika má počáteční neterminál S a terminály $x, +, *, (,)$ a $\#$:

$$\begin{aligned} S &\rightarrow E \# \\ E &\rightarrow F G \\ G &\rightarrow + F G \\ G &\rightarrow \\ F &\rightarrow T H \\ H &\rightarrow * T H \\ H &\rightarrow \\ T &\rightarrow x \\ T &\rightarrow (E) \end{aligned}$$

1. Nakreslete derivační strom pro řetězec $x + x * x \#$.
2. Určete množiny $FIRST(A)$ a $FOLLOW(A)$ pro $A \in \{E, G, F, H, T\}$.
3. Určete, zda je gramatika $LL(1)$, a stručně vysvětlete proč.

8 Algoritmy a optimalizace: hladový algoritmus (otázka pro studijní zaměření – 3 body)

- a) Na problému množinového pokrytí vysvětlete a ilustруйте použití hladových algoritmů pro NP-težké problémy.
- b) Popište instanci problému množinového pokrytí, na kterém uvedený hladový algoritmus nenajde optimální řešení.
- c) Jaký je aproximační poměr popsaného hladového algoritmu? Dokažte.

9 Algoritmy a optimalizace: vážený graf (otázka pro studijní zaměření – 3 body)

Je dán neorientovaný graf $G = (V, E)$ a celočíselná nezáporná váhová funkce w na množině $V \cup E$ (tj. každý vrchol a každá hrana má nezápornou váhu). Navrhněte jednoduchý 3-aproximační algoritmus pro problém nalezení podmnožiny vrcholů C minimalizující hodnotu

$$\sum_{v \in C} w(v) + \sum_{uv \in E, u \notin C, v \notin C} w(uv).$$

(Nápověda: použijte deterministické zaokrouhlování lineární relaxace problému.)

10 Algoritmy a optimalizace: cyklický mnohostěn (otázka pro studijní zaměření – 3 body)

Definujte cyklický mnohostěn v \mathbb{R}^d s n vrcholy. Spočítejte asymptoticky počet jeho faset (popište, které množiny vrcholů tvoří fasetu).

11 Matematická lingvistika: formální jazyky a automaty (otázka pro studijní zaměření – 3 body)

1. Vysvětlete pojem neprojektivity v závislostních stromech.
2. Uveďte příklad nějaké neprojektivní věty.

12 Matematická lingvistika: základní formalismy pro popis přirozených jazyků (otázka pro studijní zaměření – 3 body)

1. Popište základní vlastnosti teorie Funkčního generativního popisu.
2. Vysvětlete pojem valence.

13 Matematická lingvistika: základy teorie informace (otázka pro studijní zaměření – 3 body)

Pracujeme s textem v přirozeném jazyce a chceme automaticky klasifikovat slovní druhy slov bezprostředně následovaných slovesem. Na rozsáhlém vzorku dat bylo zjištěno, že

- pravděpodobnost, že nahodile vybrané slovo v textu je sloveso, je $1/8$,
- pravděpodobnost, že slovesu bezprostředně předchází příslovce, je $1/4$,
- pravděpodobnost, že slovesu bezprostředně předchází podstatné jméno, je $1/2$.

1. Vypočítejte pravděpodobnost výskytu příslovce bezprostředně následovaného slovesem.
2. Náhodná veličina reprezentující slovní druh slova bezprostředně následovaného slovesem nabývá tří různých hodnot $Y \in \{N, D, X\}$, kde N je podstatné jméno, D je příslovce a X je libovolný jiný slovní druh. Vypočítejte entropii uvedené náhodné veličiny $H(Y)$.

14 Počítačová grafika: rekurzivní sledování paprsku (otázka pro studijní zaměření – 3 body)

1. Popište princip rekurzivního sledování paprsku (ray tracing). Co je na vstupu a co je výsledkem?
2. Z jakých složek se počítá světlo při dopadu paprsku v rekurzivní proceduře `shade()`?
3. Jaké hlavní nedostatky vykazuje základní algoritmus ray tracing popsany v prvním bodě, pokud bychom ho chtěli použít jako fotorealistickou zobrazovací metodu?

15 Počítačová grafika: barvy (otázka pro studijní zaměření – 3 body)

1. Jak barvy vnímá lidské oko? Uveďte alespoň přibližný koncept.
2. Jak barvy zobrazuje počítačová technika? Popište systémy pro ukládání barev na počítači (barevné prostory).
3. Napište příklad barevného systému vhodného pro intuitivní zadávání barvy laickým uživatelem.

16 Počítačová grafika: real-time 3D grafika (otázka pro studijní zaměření – 3 body)

1. Stučně popište, jak vypadá pipeline pro vykreslení scény v existujících knihovnách pro 3D real-time grafiku s podporou HW akcelerace (např. OpenGL nebo DirectX). Zaměřte se pouze na hlavní komponenty této pipeline, které se přímo týkají renderingu scény na GPU, a stručně (nejlépe jednou větou) popište jejich funkci.
2. Která místa z výše uvedené pipeline může programátor ovlivnit (nahradit je vlastním kódem) a jak?
3. Kde je v této pipeline řešena otázka korektního zakrývání objektů (že obraz objektů blíže ke kameře může zakrýt objekty vzdálenější) a jak? Také v tomto kontextu vysvětlete, jak se zpracují poloprůhledné objekty.

17 Sítě a internetové technologie: sockety (otázka pro studijní zaměření – 3 body)

Rozhraní pro síťovou komunikaci je na řadě platform založeno na BSD socketech. Toto rozhraní je možné použít pro různé protokoly, typy adres a typy socketů. Předpokládejte, že máte k dispozici zhruba následující operace nad sockety (signatury funkcí jsou rámcové, pro připomenutí):

- `int socket(domain, sock_type)` — vytvoří socket příslušného typu pro danou doménu. Vrací file descriptor socketu.
- `int bind(socket, local_addr)` — přiřadí socketu lokální adresu. Vrací 0 v případě úspěchu (jinak kód chyby).
- `int listen(socket, backlog)` — začne čekat na příchozí spojení a nastavuje délku fronty čekajících požadavků. Vrací 0 v případě úspěchu.
- `int accept(socket, out client_addr)` — přijme spojení na socketu. Vrací 0 v případě úspěchu.
- `int connect(socket, out server_addr)` — navazuje spojení se serverem. Vrací 0 v případě úspěchu.
- `int send(socket, data)` — odešle data do socketu. Vrací počet odeslaných bajtů, jinak kód chyby.
- `int recv(socket, out data)` — přijme data ze socketu. Vrací počet přijatých bajtů, jinak kód chyby.
- `int close(socket)` — zavře file deskriptor (ukončí spojení a zruší socket).

S využitím socketového API naimplementujte:

1. Jednoduchý TCP server, který bude po navázání spojení přijímat předem neurčený počet 4-bajtových hodnot (celá čísla ve dvojkovém doplňku), a který po každém přijatém čísle vrátí jako odpověď 8-bajtové číslo (rovněž ve dvojkovém doplňku) reprezentující průběžný součet čísel dosud přijatých v rámci relace s jedním klientem.
2. Klientský program, který s pomocí výše uvedeného serveru spočte součet 10 čísel (zdroj čísel není důležitý, například náhodná čísla, čísla z příkazové řádky nebo jakkoliv jinak) a výsledek vypíše.

Předpokládejte, že klient a server mohou běžet na počítačích s odlišnými architekturami procesorů. Vaše implementace serveru nemusí obsluhovat více klientů současně.

Oba programy zapište pomocí pseudokódu na odpovídající úrovni abstrakce. V odpovědi nejsou hodnoceny drobné syntaktické chyby, pořadí nebo chybějící nepodstatné parametry funkcí.

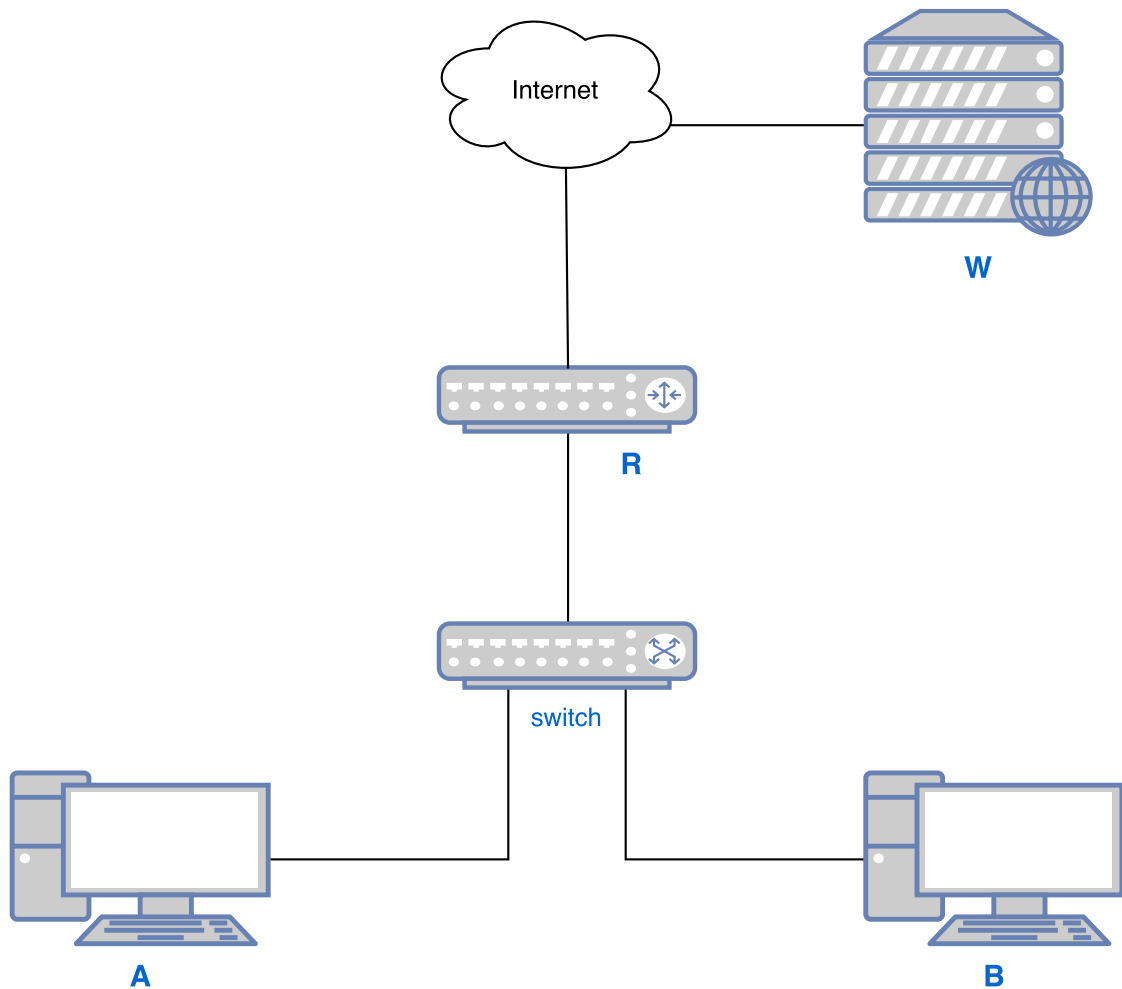
18 Sítě a internetové technologie: TCP (otázka pro studijní zaměření – 3 body)

TCP je spojovaný transportní protokol, který nad nespolehlivou sítí poskytuje spolehlivé spojení mezi dvěma koncovými body. Navíc obsahuje mechanismy pro řízení toku a ochranu proti zahlcení sítě.

1. Spolehlivost TCP je zajištěna pomocí kontinuálního potvrzování. Vysvětlete, jak kontinuální potvrzování v TCP funguje a jakým způsobem se inicializují a používají 32-bitová pole „Sequence Number“ a „Acknowledgment Number“ v TCP hlavičce.
2. Co je cílem řízení toku (flow control) při přenosu dat pomocí TCP? Vysvětlete, jak se pro tento účel používá 16-bitové pole „Window“ v TCP hlavičce.
3. Jakým způsobem může TCP detekovat zahlcení sítě (congestion) a jakým způsobem se mu snaží předcházet?

19 Sítě a internetové technologie: routing a NAT (otázka pro studijní zaměření – 3 body)

Uvažujte schéma sítě dle obrázku. V lokální síti jsou uživatelské stanice A a B, které jsou pomocí routeru R připojeny k internetu. IPv4 adresy uvnitř lokální sítě jsou z bloku privátních (neroutovaných) adres, takže při komunikaci s webovým serverem W musí router R provádět NAT.



1. Přidělte všem rozhraním konkrétní IPv4 adresy.
2. V jednotlivých krocích popište přenos jednoho IPv4 datagramu mezi počítači A a B. Předpokládejte, že počítače ani router před zahájením přenosu nemají žádné ARP informace.
3. V jednotlivých krocích popište odeslání jednoho IPv4 datagramu a přijetí jednoho IPv4 datagramu v rámci TCP spojení mezi počítačem A a webovým serverem W v internetu. Předpokládejte, že počítače mají všechny potřebné ARP informace a popisujte pouze akce počítače A, routeru R a webového serveru W. Detaily TCP protokolu popisujte pouze do té míry, do které je to nutné pro NAT.

20 Systémové programování: rozhraní pro synchronizaci (otázka pro studijní zaměření – 3 body)

Jedním ze základních synchronizačních objektů je semafor (semaphore). Implementuje jej například Linux:

```
int sem_init (sem_t *sem, int pshared, unsigned int value);
int sem_destroy (sem_t *sem);
int sem_post (sem_t *sem);
int sem_wait (sem_t *sem);
int sem_trywait (sem_t *sem);
```

Nebo Windows:

```
HANDLE WINAPI CreateSemaphore (
    _In_opt_ LPSECURITY_ATTRIBUTES lpSemaphoreAttributes,
    _In_     LONG                   lInitialCount,
    _In_     LONG                   lMaximumCount,
    _In_opt_ LPCTSTR               lpName
);
```

```
DWORD WINAPI WaitForSingleObject (
    _In_ HANDLE hHandle,
    _In_ DWORD dwMilliseconds
);
```

```

BOOL WINAPI ReleaseSemaphore (
    _In_      HANDLE hSemaphore,
    _In_      LONG   lReleaseCount,
    _Out_opt_ LPLONG lpPreviousCount
);

```

```

BOOL WINAPI CloseHandle (
    _In_ HANDLE hObject
);

```

1. Pro jednu z uvedených implementací popište semantiku operací včetně argumentů (ideálně všech argumentů, ale méně zásadní argumenty je možné vynechat). Jako nouzovou alternativu navrhněte rozhraní vlastní implementace a popište jeho semantiku.
2. Načrtněte kód, ve kterém jedno vlákno umístí do sdílené proměnné celočíselného typu hodnotu, na kterou druhé vlákno čeká. Čekání implementujte pomocí semaforu.

21 Systémové programování: alokátory (otázka pro studijní zaměření – 3 body)

Jako alokátor zde označujeme implementaci dynamické alokace a uvolňování paměti na heapu, dostupnou například funkcemi `malloc` a `free`, případně operátory `new` a `delete` a podobně. Alokátor může být součástí základních runtime knihoven operačního systému, případně prostředí daného programovacího jazyka.

1. Uvažujte situaci, kdy můžete data alokovat jak na zásobníku, tak na heapu (jde o lokální alokaci uvnitř funkce a data neopustí hranice funkce). Jakou alokaci byste preferovali pro typ „integer“ a typ „pole integerů velikosti 1000000 položek“ a proč?
2. Na příkladu sekvence volání alokátoru ilustруйте, jak může vzniknout fragmentace heapu.

22 Systémové programování: interní struktura základních systémů souborů (otázka pro studijní zaměření – 3 body)

Uvažujte systém souborů FAT v nějaké základní verzi, například FAT 16 bez podpory dlouhých jmen souborů.

1. Vysvětlíte, v jakých jednotkách přiděluje tento systém souborů diskový prostor. Napište velikost diskového prostoru obsazeného daty souboru dlouhého 0 B, 1 B, 1 MiB.
2. Napište, jak je v tomto systému souborů uložena struktura adresářů (stačí popis celkového konceptu a seznam hlavních polí adresářové položky). Z vašeho popisu určete, kolik přístupů na disk musí vykonat operace nalezení prvního bajtu souboru podle plného jména s cestou (určete parametry, na kterých tento počet záleží, a napište odhad ve stylu výpočetní složitosti, nemusí jít o přesné číslo).

Jako nouzovou alternativu můžete tuto otázku zodpovědět pro jiný přiměřeně rozšířený systém souborů.

23 LA1 – Lineární zobrazení (3 body)

Pro lineární zobrazení $f: x \mapsto Ax$ s maticí $A \in \mathbb{R}^{m \times n}$ vyjádřete vzorečkem vztah mezi

- a) hodnotí matice A a dimenzí obrazu $f(\mathbb{R}^n)$,
- b) hodnotí matice A a dimenzí jádra.

24 LA2 – Determinant (3 body)

Definujte pojem determinant matice. Jak lze pomocí $\det(A)$ určit determinant matice inverzní k matici A ? Zdůvodněte.

25 LA3 – Norma (3 body)

Rozhodněte, zda $\|A\| := \max_{i=1, \dots, m} \sum_{j=1}^n |a_{ij}|$ je normou na prostoru matic $A \in \mathbb{R}^{m \times n}$.

26 MA1 – Řady (3 body)

Definujte součet nekonečné řady, formulujte nutnou podmínku konvergence a uveďte příklad, kdy není postačující.

27 MA2 – Plocha (3 body)

Spočtete plochu ohraničenou grafy funkcí $5 - x^2$ a $2x + 2$.

28 Pravděpodobnost (3 body)

Mezi všemi 3-prvkovými podmnožinami množiny $\{0, \dots, 10\}$ si jednu zvolme náhodně (každou se stejnou pravděpodobností), a označme si s její součet. Jaká je střední hodnota s ? Dokažte, že pravděpodobnost, že $s \geq 20$, je nejvýše $3/4$.

29 Grafy (3 body)

Nechť G je souvislý graf s alespoň dvěma vrcholy. Ukažte, že G obsahuje alespoň dva vrcholy v_1, v_2 takové, že $G - v_i$ je souvislý pro $i = 1, 2$. Které grafy obsahují právě dva takové vrcholy?

30 Kombinatorika (3 body)

Nechť $N = 3 \cdot 5 \cdot 7$. Kolik čísel menších než N je s N nesoudělných?

31 Logika (3 body)

Před soudem stojí dva obžalovaní, pan A a pan B.

- (i) Pan A říká: "Jsem nevinen nebo pan B lže."
- (ii) Pan B říká: "Pan A je vinen nebo lžu."
- (iii) Právě jeden obžalovaný je vinen.

Nechť prvovýroky v_A, v_B reprezentují, že pan A (resp. pan B) je vinen a p_A, p_B reprezentují, že pan A (resp. pan B) říká pravdu. Označme $\mathbb{P} = \{v_A, v_B, p_A, p_B\}$.

1. Napište výroky φ_1, φ_2 ve tvaru ekvivalence (tj. pomocí logické spojky \leftrightarrow mezi nějakými podvýroky) a výrok φ_3 v CNF vyjadřující (po řadě) (i), (ii), (iii), vše nad množinou prvovýroků \mathbb{P} .
2. Pomocí nějaké formální dokazovací metody (tablo metoda, rezoluce, Hilbertovský kalkul) dokažte, že z $T = \{\varphi_1, \varphi_2, \varphi_3\}$ vyplývá, že pan B je nevinen, tj. $T \models \neg v_B$.
3. Uveďte definici pojmu *výrok nezávislý v teorii T*. Napište příklad výroku nad \mathbb{P} , který je nezávislý ve výše uvedené teorii T , anebo zdůvodněte, proč takový výrok neexistuje.